SYLLABUS

Basic data of the course:	
Academic unit:	**Faculty of Engineering and Informatics**
Course title:	**Programming**
Level:	**Bachelor**
Course status:	**Core**
Year of studies:	**II**
Number of hours per week:	**3**
Value in credit – ECTS:	**5**
Time / location:	**Monday, 13: 00-16: 00, Room 203**
Course teacher:	**Prof. Ass. Dr. Fakije Zejnullahu**
Contact details:	fakije.zejnullahu**@ushaf.net**
Course description:	*This course will enables students to apply programming techniques to new software projects. This course also enables students to successfully learn and apply object programming concepts and techniques. Also this course will introduce and enable students to apply object-oriented programming techniques to software.*
Aim of the course:	*The aim of the course is to equip students with modern knowledge in "thinking and programming", a prerequisite for the basics of programming. In addition, students in this course will learn to program with strings and matrices in the c # programming language. Familiarizing students with algorithms and their presentation forms. Students will gain knowledge of the concept of computer programming, utilizing the C # programming language as the main development tool, using C # algorithms and programming language.* *Requirements for completing the goal of this course are:* • *Programming skills* • *Active student during lectures and exercises.*
Expected outcomes from learning:	*After completing this course (subject) the student will be able to:* • *analyze and solve the problem,* • *use C # programming language to solve the problem,* • *understand the key concepts of object-oriented programming,* • *be able to write class code and use objects,* • *implement inheritance and polymorphism in code,* • *be able to handle mistakes,* • *identify the complexity of programming problem solving methodologies.*

Student contribution (which should correspond to the student's learning)			
Activity	**Hour**	**Day / week**	**Total**
Lectures with numerical exercises	3	15	45
Internship			
Contacts with teacher / consultations	1	3	3
Field exercises			
Midterm, seminars and projects.		10	10
Homework			
Self-learning time student (at the library or at home)	3	15	45
Final preparation for the exam	2	8	16
Time spent on evaluation (tests, quiz and final exam)	1	3	3
Projects and presentations.	1	4	4
Total			**126**

Teaching methodology:	*Lectures and exercises combined with case studies and classroom discussions*
Evaluation methods:	*The student can choose to be evaluated one of the two forms of evaluation, given below:* *1. Form 1: Assessment with colloquiums and project 2. Form 2: Assessment with the final exam.* *Form 1:* *In the first form of assessment "Assessment with colloquiums and project" the student is assessed in four activities that are carried out during the lectures:* *1. Colloquium 1 (30%), individual assessment* *2. Colloquium 2 (30%), individual evaluation* *3. Class activity (10%), individual assessment* *4. Project (30%), group assessment.* *Additional clarification:* *If the student in each activity above reaches the maximum points, then he will be evaluated with 100 points.* *Students who pass the exam according to form 1 of the assessment, are released from the obligation to take the final exam. Only if the student is not satisfied with the grade achieved according to form 1, then he can undergo the final exam to obtain a higher grade.* *Form 2:* *In the second form of evaluation, "Evaluation with the final exam", the student will undergo the exam which is held after the completion of the course lectures, and is organized in the exam deadlines, determined by the University senate. Through the final exam, the student can achieve a maximum of 70% of the points from the total of 100 points.* *The rest of the 30% points must be completed by group work in the Project, an activity carried out during the*

	lectures. *In Colloquium 1, Colloquium 2 and the final exam, the evaluation of students will be done through an evaluation form,* *which must be completed individually by the student. The evaluation form will contain objective and subjective questions through which the student's learning outcomes will be evaluated:* *• The objective questions will be of the following types: (1) Multiple choice questions, (2) True/False, (3) Completion and (4) Composition/Matching; questions that will be used to assess the student's ability to recall and recognize concepts and course material.* *• The subjective questions will be of the Essay/written task type that will be used to assess the student's understanding and abilities to apply the knowledge gained in the analysis, synthesis and evaluation of the problem, from the answers prepared by the student to the question of submitted.* *Activity in the class means the student's engagement in dealing with the issues discussed in the class, during the lectures.* *Project (30%), group assessment: it is an activity in which students apply the acquired knowledge in a concrete project. It is carried out in groups of 2 or 3 students who are obliged to carry out the activity, document and present it to the subject professor.* *For the form of realism and documentation of the activity, all members of the group will be evaluated with the same point (20%), while the evaluation of the presentation skills of the activity is individual and includes 10%.* *Rating:* *91-100 points - graded 10 (ten)* *81-90 points - graded 9 (nine)* *71-80 points - grade 8 (eight)* *61-70 points - graded 7 (seven)* *51-60 points - grade 6 (Six)* *0-50 points – The student repeats the exam.*
Literature	
Basic literature:	*1. C# Programming: From Problem Analysis To Program Design-Barbara Doyle* *2. Robert Harle, "Object Oriented Programming", IA NST CS and CST Lent 2009/10*
Additional literature:	*-Troelsen, A., & Japikse, P. (2017). Pro C# 7: With. NET and. NET Core. Apress.* *-Libra online: https://introprogramming.info/english-intro-csharp-book/*
Designed lesson plan :	
Week	**The lecture to be held**

Week one:	Introduction to programming language and introduction to the course
Week two:	Loop Loop: while, do-while, for. Algorithms and loop programming. Reduce numerical errors.
Week three:	Methods (functions)
Week four:	Types of methods (functions)
Week five:	Arrays
Week six:	*First Evaluation*
Week seven:	Exceptions and error handling
Week eight:	Basic classes, static and partial
Week nine:	Objects, Constructors and destructors
Week ten:	Data access, attributes, properties, and methods
Week eleven:	Inheritance and polymorphism of classes
Week twelve:	Abstract classes and interfaces
Week thirteen:	Basics of Graphical User Interface
Week fourteen:	Study visits to a company
Week fifteen:	*Second evaluation*
Academic policies and rules of conduct	
Regular attendance of lectures and exercises is necessary, as well as active participation with discussion and solution of tasks. Not impeding the progress required for learning using mobile phones turned off or in silent mode.	